2. 画像処理技術を利用した携帯電話向けアプリケーションの現状と今後

津田 雅之[†], 磯 俊樹^{††}

キーワード ● カメラ付き携帯電話機, Java, ID読取, Visual tag, 画像処理

1. ま え が き

電気通信事業社協会の調べによると、日本国内の携帯電話の契約者数(累計)は、2004年2月末現在で80,541,500台となった。このうち、カメラ付き携帯電話の数は約4,605万台であり、現状、国内の携帯電話の半数以上がカメラ付き携帯電話ということになる。数年後には、8割から9割の携帯電話がカメラ付きになっているものと予想されている。

急速に普及しているカメラ付き携帯電話であるが、その主な用途は、デジタルカメラとしての静止画の撮影、デジタルビデオカメラとしての動画の撮影である。撮影された画像の利用としては、メールへの添付やメモリーカードへの保存、赤外線通信を使ったプリンタへの出力など、撮影された画像をそのまま利用することが多い。すなわち、デジタルカメラやデジタルビデオカメラの代用として、携帯電話のカメラ機能が利用されている。その結果、撮影画像の画質に対するユーザの要望が次第に高くなり、最初のカメラ付き携帯電話(2000年11月発売のJフォン(現、ボーダフォン)J-SH04)では、カメラには11万画素のCMOSセンサが使用されていたのに対して、表1に示すように2004年3月末の最新機種では、最高202万画素のCCDが使われるようになった。また、CCDのサイズも1/2.7型から1/3.8型が利用されており、3年前のカメラ付き携帯電話で撮影した画像に比べて、高い画質の画像が撮影できるようになった。

カメラ付き携帯電話の普及と性能向上により、デジタルカメラやデジタルビデオの代用としての利用だけでなく、画像処理によるモバイル向けのサービスを実現しようとする動きが出てきた。携帯電話は、PCやワークステーションとは異なり、組込み機器であること、また、CPUやメモリーが非力であることから、画像処理を実現する方法に制限がある。現状、カメラ付き携帯電話を利用した画像処理を実現する方法としては、組込みソフトウェアとして画像処理を実装する方法と、ダウンロードアプリケーションとして画像処理を実装する方法に大別される。前者は、携帯電話のROMにソフトウェアを焼き付けたり、専用ハードウェアを携帯電話に実装することで画像処理を実現するものである。実際に携帯電話上で実装されている画像処理としては、バーコードやOCR、2次元コードの認識、エッジ強調、色調の変更、逆光修正などがある。後者は、インターネットからダウンロードしたアプリケーションで画像処理を実現するものである。約4年前から携帯電話上でも、JavaやC++でプログラムされたアプリケーションがダウンロードでき、それを実行できるようになった。組込みソフトウェアによる画像処理は処理の高速化が可能である反面、バージョンアップ等のカスタマイズに限界がある。それに対して、ダウンロードアプリケーションによる画像処理は、初めにダウンロードする必要性がある反面、柔軟なカスタマイズが可能という特徴を持つ。さらに、ソフトウェア形式で実現する画像処理は、

表1 第3世代携帯電話 最新機種(2004.3現在)のカメラ機能一覧

機 種	NTTドコモ SH900i	NTTドコモ F900i	NTTドコモ N900i	NTTドコモ P900i
撮像素子	CCD	CCD	スーパーCCDハニカム	CCD
撮像素子サイズ	1/2.7型	1/2.7型	1/3.8型	1/3.6型
カメラ有効画素数	202万画素	128万画素	100万画素	128万画素
静止画記録サイズ(ドット数)	240×320, 352×288 480×640, 768×1024 960×1280, 1224×1632	240×320, 352×288 480×640, 640×480 960×1280	240×269, 352×288 640×480, 1280×960 1616×1212	128×96, 176×144 240×266, 352×288 640×480, 1280×960

[†] 株式会社NTTドコモ 移動機開発部

^{††} 株式会社NTTドコモ ネットワーク研究所

"Present and Future of Cell Phone Applications Using Image Processing" by Masayuki Tsuda (NTT DoCoMo Customer Equipment Development Department, Tokyo) and Toshiki Iso (NTT DoCoMo Network Laboratories, Tokyo)

- (1) サーバ型画像処理：画像処理をサーバ上で実行
- (2) クライアント型画像処理：画像処理を携帯電話上で実行
- (3) 複合型画像処理：サーバならびに携帯電話上で実行に分けられる。サーバ型は、撮影した画像をサーバ側へ送信し、サーバ上での処理結果を携帯電話に返すことでサービスを実現している。画像の送信や処理結果の受信には、ダウンロードアプリケーションが、そのプログラムの中でHTTP通信の機能呼び出すことで実現している。一方、クライアント型は、携帯電話のCPUの非力さやメモリーの少なさをさまざまな工夫により克服し実現している。また、複合型は、サーバと携帯電話機で、それぞれユーザビリティや通信コストを考慮した分散処理により実現している。

本稿では、上記の組込みソフトウェアによる画像処理と、ダウンロードアプリケーションでサーバ型ならびに複合型画像処理を使った携帯電話向けサービス事例ならびに実装概要を中心に解説する。

2. 組込みソフトウェアによる画像処理実現例

携帯電話にROM化されている画像処理ソフトウェアとしてはコード認識がある。2004年3月現在、携帯電話のカメラを利用して認識させることが可能なコードとしては、バーコードや2次元コードがある。特に2次元コードとしては、デンソーが開発したマトリックス型2次元コードQuick Responseコード（QRコード）¹⁾が、NTTドコモ、ボーダフォン、KDDIの3社で採用されており、広く普及している。以下ではQRコードについて解説する。

QRコードは、主に文字入力の手代手段として活用されている。例えば、名刺にQRコードを印刷しておけば、これをカメラ付き携帯電話で読取ることで、名前や電話番号、Eメールアドレスなどが、手間をかけずに携帯電話の電話帳に保存できるようになる。また、カタログショッピング雑誌に掲載されている商品の横に印刷されたQRコードをカメラ付き携帯電話で読取ることで、インターネット上の商品注文サーバへアクセスするURLが取得でき、そのURLへ携帯電話のブラウザを使ってアクセスし商品注文することができる。どちらの場合にも、親指とキーボードによって行っていた文字入力手段を、カメラと認識エンジンで代替するのである。このようなQRコードの使い方は、ゆっくりと拡がりを見せ始めている。グルメなどのタウン情報サイトのURLを埋め込んだQRコードが印刷された書籍地図帳が販売されていたり²⁾、着メロサイトのURLを埋め込んだQRコードが印刷されたポスターなどが駅などに掲示されていたりしている。

QRコードには、モデル1、モデル2、マイクロQRコードの3種類がある¹⁾。現状、カメラ付き携帯電話で認識できるQRコードはモデル2であり、QRコードの大きさを表すバージョンには1から10に対応している。なお、バージョン

が一つあがるごとにQRコードは4セル分大きくなる。QRコードのモデル1、モデル2では、三つのコーナに切出しシンボルと呼ばれる位置検出用パターンがあり、カメラ付き携帯電話でQRコードを撮影する際にこのパターンを自動的に検出し、撮影を行い、そしてQRコードの切出しを行っている。誤り訂正は4段階（レベルL、M、Q、H）あり、それぞれQRコードの面積の7%、15%、25%、30%が汚れたり、破損した場合でもデータを復元することができる。カメラ付き携帯電話では、レベルHまでサポートしている。コードの読取り条件としては、NTTドコモの場合、セルサイズ0.28mm以上、読取り距離5から10cm、コントラスト55%、白色度85%以上の上質紙を規定している。なお、読取り距離が短いため、携帯電話のカメラ機能として接写機能が標準に搭載されている。

QRコード以外の2次元コードとしては、オリンパスが開発したSTコードがある³⁾。ただし、STコードの認識エンジンが組込みソフトウェアとして携帯電話に実装されている例はまだなく、KDDI社の携帯電話向けのダウンロードアプリケーションとしての実装例があるだけである。STコードを用いたサービスとしては、新聞に印刷されたSTコードを認識することで、記事を音声で読み上げるサービスなどがある。

3. ダウンロードアプリケーションの開発環境

携帯電話上へダウンロードして、実行可能なアプリケーションを開発するための環境として、ここではプログラミング言語について説明する。現状では、開発用言語としてJavaとBrew上でのC++がある。ここでは、筆者らが精通しているJava、特にNTTドコモで開発をしているJava（DoJaと呼ぶ）⁴⁾について解説する。Brewについては、文献5)を参照されたい。

携帯電話向けのJavaはJ2ME（Java 2 Micro Edition）と呼ばれ、PCなどで利用されているJava（J2SE（Java 2 Standard Edition））のサブセットとなっている。その構成は、VM（Virtual Machine）、CLDC（Connected Limited Device Configuration）と呼ばれている基本クラス群、プロファイルと呼ばれる携帯電話の特長を活かすクラス群から成る。プロファイルは多数存在し、Javaの仕様を標準化しているJCP（Java Community Process）で規定されたプロファイルやDoJaのように通信事業者が規定したプロファイルなどがある。NTTドコモが開発しているDoJaプロファイルを用いて携帯電話上で画像処理を行う場合、

- (1) com.nttdocomo.device.Cameraクラス
- (2) com.nttdocomo.device.CodeReaderクラス
- (3) com.nttdocomo.uiクラス
- (4) com.nttdocomo.HttpConnectionインタフェース

が主に用いられる。以下に各クラスやインタフェースの簡単な概要と画像処理を行う場合の流れを示す。

Cameraクラスは、携帯電話のカメラ機能をJavaから利用するためのAPI群がまとめられたクラスである。このクラスを利用して静止画像を撮影するには、まず、メソッドgetCamera()を呼び出し、カメラオブジェクトを取得する。次に、撮影する画像のサイズをsetImageSize()で、画質をメソッドsetAttribute()で、取得したカメラオブジェクトに対して指定する。そして、カメラオブジェクトに対してtakePicture()を呼び出すことで、携帯電話のカメラ機能が呼び出され、画像が指定したサイズや画質で撮影される。撮影された画像はカメラオブジェクトに取り込まれる。カメラオブジェクトから撮影した画像をJavaが扱えるデータとして取出すには、メソッドgetImage()やメソッドgetInputStream()を用いる。

1章でも述べたように、携帯電話のダウンロードアプリケーションとして画像処理を行う場合には、サーバで画像処理を行う場合と携帯電話の中で画像処理を行う場合がある。サーバで画像処理を行う場合、Cameraクラスを用いて撮影し、取得した画像をHTTP/HTTPS通信によりサーバに送信する。現状の携帯電話にはサーバからのプッシュによる通知取得機能がないため、サーバ側で処理した結果は、HttpConnectionオブジェクトのGETによるプル型通信で取得するか、メールで通知し、そのメールから処理結果をJavaで取得するしかない。

携帯電話の中で画像処理を行うには、カメラオブジェクトのメソッドgetInputStream()で取得したデータを直接処理するか、uiクラスを利用する。uiクラスを利用する場合、カメラオブジェクトのメソッドgetImage()を用いて、MediaImageオブジェクトに変換し、そして、MediaImageクラスgetImage()によりイメージオブジェクトを取得する。このイメージオブジェクトを、ImageクラスのcreateImage()で生成したイメージオブジェクトもしくはCanvasクラスのキャンバスオブジェクトに一旦、描画する。カメラで撮影した画像を描画した後、GraphicsクラスのgetRGBPixel()もしくはgetRGBPixels()を用いて、撮影した画像の画素ごとの色を0xRRGGBB形式で取得する。そして、画像処理が行われる。なお、2004年3月現在の携帯電話に実装されているCLDCバージョン1.0には浮動小数点がサポートされていないため、携帯電話の中の画像処理での小数点演算は、アプリケーションごとに計算ルーチンが実装され、利用されている。CLDCの次期バージョンであるCLDCバージョン1.1では、J2SE同様の浮動小数点がサポートされている。

DoJaでは、携帯電話に実装されているコード認識エンジンをJavaから呼び出すことができる。CodeReaderクラスのメソッドgetCodeReader()を用いて、コードリーダーオブジェクトを取得する。そして、このコードリーダーオブジェクトに認識させるコードの種類(バーコード(JAN8, JAN13), 2次元コード, OCR)を設定する。次に、コード

リーダーオブジェクトに対してメソッドread()を実行することで、携帯電話のカメラでコードを撮影し、認識エンジンで処理し、認識結果をコードリーダーオブジェクトに返す。コードリーダーオブジェクトからデータを取得するには、getString()やgetBytes()を用いる。getString()により文字列として、getBytes()によりバイト列として認識結果を取得する。

DoJaでのカメラ機能のユースケースは、撮影した画像とキャラクタ等のフレーム画像の合成が主たるものであり、それ以外の画像処理はあまり想定されていない。このことは、Javaの標準仕様であるマルチメディアAPIでも同様である。このため、特に、画像処理をダウンロードアプリケーションとして携帯電話上で実行する場合に、制限が存在することは不可避となっている。

4. ダウンロードアプリケーションとしての画像処理実現例

ここでは、携帯電話機を使って容易に情報やサービスにアクセスするための基盤技術として、複合型とサーバ型の画像処理によるID読取技術を例に取って紹介する。

4.1 電子透かしによるID読取

デジタルコンテンツに埋め込まれた電子透かしの著作権情報を、そのコンテンツに関連したサービスへ仲介するための起点情報として利用するシステムがある(図1)^{6) 7)}。これは、電子透かしが埋め込まれたデジタルコンテンツは勿論のこと、アナログ出力された印刷物であっても、それを携帯電話機に搭載されたカメラで読み込むことで、そのコンテンツに関連付けられたコンテンツ配信やオンラインショップ、チケット予約といったサービスへ誘導することができるシステムである。

これらのサービスを実現するためには、以下の要件を満

図1 電子透かし技術を利用したサービス仲介システム (文献7)より引用)

たす必要がある。

- (1) 幾何歪みに対するロバスト性
- (2) ユーザビリティを考慮したレスポンス応答
- (3) 処理コストならびに処理時間を考慮した通信量

(1) は、基本的に電子透かしアルゴリズムが幾何歪みには弱いことを意味している。(2)、(3) は、ユーザビリティの観点から、電子透かしの検出の可否を極力早くユーザへ通知することの必要性を意味している。これらの要件を、上記のシステムでは複合型画像処理により実現している(図2)⁷⁾。すなわち、クライアントである携帯電話機上では、電子透かしの検出可否処理までを、サーバ上では、電子透かしの解析ならびに結果通知処理までを担当する。そして、(1)を満たすために、コンテンツの四隅に配置された幾何補正用マークをパターンマッチング処理により抽出し、それから算出された幾何変換パラメータに基づき幾何補正処理する。この処理により、読取角度は x, y 方向それぞれ 15° までであれば(1)を満足させることができる。また、(2)、(3)を満たすために、前処理で得られた幾何変換パラメータに基づき電子透かし抽出可否を判断する処理や、電子透かし技術が画像データを小さな輝度ブロックに縮退して検出する処理を施している。

4.2 VisuالتagによるID読取

Visuالتag (図3) によるID読取システム⁸⁾とは、街中にある広告やポスターからCM映像情報や、観光案内板から行き先のライブ映像などを携帯電話機で手軽に見るためのシステムである⁹⁾。このシステムは、いつでもどこでも現実世界で目にしたモノのIDから関連するサービスが利用できることを目指している。その特長としては、IDに関連する映像情報をダウンロードすることなくリアルタイムで閲覧できるため、次々とIDタグを読み込むことが可能となる点である。また、IDタグを正面からや、近接距離からでなくと

もIDを読取ることができる点もその特長の一つである。これらの特長を実現するためには、以下の要件を満たす必要がある。

- (1) ID読取とサービス提供のリアルタイム処理
- (2) ID読取条件に対するロバスト性

(1) は、ID読取とサービス提供を同時に処理することを意味する。また、(2) は、バーコードや2次元コードのように、読取りの際に正対させ、ほぼ接写させる必要があるID読取技術とは違い、読取角度や読取距離にロバストであり、さらには、照明条件にもロバストなID読取技術であることを意味する。これら要件を、VisuالتagによるID読取システムでは、サーバ型画像処理を用いて実現している。具体的には、(1)を満たすために、携帯電話とシステムとの間をIMT-2000携帯電話（いわゆる第3世代携帯電話）のテレビ電話モードで接続し、その上り回線をIDタグ映像の連続的な送信に、システムサーバをID読取処理に、さらに下り回線を読取ったIDに関連する映像送信にそれぞれ用いている(図4)。また、(2)を満たすために、視点に依らない不変量（幾何学的不変量）に基づいて生成されたIDタグであるVisuالتag⁹⁾を用いるとともにモバイル環境下でもロバストな画像処理技術を利用している。

サーバ型画像処理としてVisuالتag読取技術の概要を簡単に述べる。処理は大別して、画像取込処理、Visuالتag領域抽出処理、特徴点位置抽出処理、ID抽出処理からなる。これら処理のうち、モバイル環境下でサービスを実現するために、照明条件にロバストな画像処理をしているVisuالتag領域抽出処理と特徴点位置抽出処理が重要となる。Visuالتag領域抽出処理では、Visuالتag領域の境界を表す円形マーカと境界線のエッジ情報を抽出し、それぞれの配置の整合性を考慮することで、Visuالتag領域を抽出している。このように、境界領域を表す異なるマーカ（円形マー

図4 Visuالتag読取技術を利用したユビキタスサービスシステム

カと境界線) 同士の整合性を考慮することで、どちらかのマーカの一部分が撮影時の手影等で照明条件が局所的に変わってもロバストなVisuالتag領域抽出が可能である。また、特徴点位置抽出処理では、前段で抽出されたVisuالتag領域を局所領域に分割し、それぞれの領域ごとにVisuالتagの背景色が白色になるような色補正処理を施すことで特徴点位置を抽出している。このように局所領域ごとの色補正処理により、照明条件にロバストな特徴点位置抽出が可能となっている。

5. 今後の展開

今後、画像処理を使った認証やコミュニケーション支援などが期待されている。特に、認証技術については、携帯電話に搭載された非接触ICを用いたモバイルEコマースの普及を控え、携帯電話自体やサービス自体のセキュリティを高める手段として期待されている。携帯電話のカメラを用いた認証技術として期待されているものとしては、顔画像認証や虹彩認証などが挙げられる¹⁰⁾。これら認証技術の利用用途を考えると、携帯電話が圏外となった場合でも認証技術が使える必要がある。指紋認証センサがチップ化されて携帯電話に搭載されているのと同様に、顔画像認証や虹彩認証についてもチップ化もしくはROM化されて携帯電話に搭載されることが望まれる。コミュニケーション支援としての画像処理への期待は、テレビ電話機能を利用したアバター通信が携帯電話上でサービスされていることによる¹¹⁾。また、3Dグラフィックスを用いた擬人化エージェントソフトウェアのサービスも行われている。これらサービスにおいて、撮影した顔画像から特徴点を抽出し、利用者の顔の向きなどをリアルタイムでトラッキングできれば、

アバターや擬人化エージェントの表現力を向上させ、通信相手に細かな雰囲気を与えることが可能となる¹²⁾。

6. むすび

画像処理を使った現状での携帯電話向けサービス事例ならびに実装概要について述べた。

携帯電話機は、この数年での急速な普及と高性能化・多機能化により、単なるコミュニケーションツールの枠を越え、人とサービスをつなぐための“サービスゲートウェイ”としての利用に期待が高まっている。そのような中で、画像という最も情報量を有するメディアを処理する技術が、今後の携帯電話機で利用できるサービスの多様化をさらに加速させるものと期待される。

(2004年4月5日受付)

〔文 献〕

- 1) 平本：“知っておきたいバーコード・2次元コードの知識”，日本工業出版（2001）
- 2) アルプス社：“携帯mini東京マップ”（2004）
- 3) STコード：“<http://www.olympus.co.jp/jp/info/2004a/if040224stcodej.cfm>”（2004）
- 4) R. Barbagall：“Wireless Game Development in C/C++with Brew”，Nat'l Book Network（2003）
- 5) NTTドコモ：“iアプリコンテンツ開発ガイド for DoJa-3.0詳細編”，http://www.nttdocomo.co.jp/p_s/imode/java/index.html（2003）
- 6) 片山ほか：“コンテンツを起点に電子商取引に誘導するサービス仲介ゲートウェイ”，NTT技術ジャーナル，14，10（2002）
- 7) 中村ほか：“カメラ付き携帯電話機を用いたサービス仲介システムのための電子透かし検出方式”，FIT，N-020（2003）
- 8) T.Iso et al.：“Visual-Tag Reader：Image Capture by Cell Phone Camera”，IEEE ICIP'03，3，pp.557-560（2003）
- 9) T.Iso et al.：“Ubiquitous Video Streaming System with a Cell Phone Camera”，IEEE PACRIM'03，1，pp.189-192（2003）
- 10) オムロン，<http://www.face-id.jp/s08-mbl/s08.html>
- 11) キャラ電（NTTドコモ），<http://900i.nttdocomo.co.jp/main.html/main.html>
- 12) アイマティックジャパン社，http://www.eyematic.co.jp/img/nikkeisangyo_030703.pdf

津田 雅之 1992年，早稲田大学大学院理工学研究科修士課程修了。同年，日本電信電話（株）入社。NTTヒューマンインタフェース研究所にて，ロボットの知的プログラミングおよびコントロールシステムの研究に従事。1999年，NTT移動体通信網株式会社（現，株式会社NTTドコモ）に転籍。現在，移動体開発部にて，携帯電話向けJava機能の開発に従事。博士（工学）。

磯 俊樹 1988年，早稲田大学大学院修士理工学研究科修士課程修了。同年，日本電信電話（株）入社。NTTヒューマンインタフェース研究所，ATR人間情報通信研究所を経て，2000年より，NTTドコモマルチメディア研究所に勤務。画像認識，ユビキタスコンピューティングの研究に従事。現在，NTTドコモネットワーク研究所にて，知的センサネットワーク処理の研究に従事。